

En esta entrada vamos a presentar un poco el framework SpringBoot2 para Java, es un software que como objetivo tiende a sustituir y eliminar los tediosos archivos *XML* del framework de Spring por las *anotaciones*, que básicamente son clases programadas que modifican el comportamiento de las clases, métodos y parámetros sobre los que están puestos, y se distinguen en el código por tener una @ delante del nombre de la anotación.

Tenemos decenas de *anotaciones* en el framework que nos van a ayudar a la hora de implementar un proyecto de forma muy rápida sin tener que reescribir el código necesario para que todo funcione, parece *magia* ya que todo es transparente para el programador, aunque nos obliga a formarnos, tiene una curva de aprendizaje muy rápida y sus funcionalidades pueden ayudarnos a la hora de integrar la lógica de la aplicación.

## @anotaciones

```
@Autowired // Instancia una variable y sus @Beans
@Bean // Crea una instancia singleton
@SpringBootApplication // Instancia una aplicación SpringBoot2
@Controller // Clase controlador
@Service // Clase servicio
@Repository // Clase repositorio
@SpringBootTest // Instancia una bateria de pruebas
@RequestMapping // Prepara un método para un endpoint
@GetMapping // Mismo que el anterior HTTP GET
@PostMapping // Mismo que el anterior HTTP POST
@Async // Clase asíncrona
@EnableAsync // Instancia una aplicación asíncrona
```

Las anotaciones son muy potentes, nos permiten añadir funcionalidad al código sin tener que

implementarla, podemos pasar parámetros de entrada en la anotación entre paréntesis () o varios entre llaves ({}), eso dependerá según para lo que esté programada la anotación.

```
@GetMapping({ "/", "/index" }) // dos parámetros
@PostMapping("/new") // un parámetro
@GetMapping("/{id}/edit") // podemos especificar parámetros
enlazados con variables de entrada
```

Como vemos con estas herramientas fácilmente podríamos implementar una aplicación con el patrón MVC y añadir las funcionalidades necesarias como un simple CRUD

## Spring Initializr para SpringBoot2

Para iniciar una aplicación en SpringBoot2 vamos a la web [start.spring.io](http://start.spring.io) y seleccionamos el tipo de proyecto que vamos a desplegar y las librerías que vamos a utilizar

The screenshot shows the Spring Initializr web application. The interface is divided into several sections:

- Project:** Maven Project (selected) and Gradle Project.
- Language:** Java (selected), Kotlin, and Groovy.
- Spring Boot:** 2.2.0 M6, 2.2.0 (SNAPSHOT), 2.1.9 (SNAPSHOT), and 2.1.8 (selected).
- Project Metadata:** Group: com.example, Artifact: demo, and a section for Options.
- Dependencies:** A search bar with the text "Web, Security, JPA, Actuator, Devtools..." and a list of selected dependencies: Lombok and Spring Reactive Web. A "5 selected" badge is visible in the top right of the dependencies section.

At the bottom, there are two buttons: "Generate the project - Ctrl + G" and "Explore the project - Ctrl + Space".

© 2013-2019 Pivotal Software  
start.spring.io is powered by  
[Spring Initializr](#) and [Pivotal Web Services](#)

Una vez terminamos nos descargamos el proyecto y lo *descomprimos* en el workspace para empezar a trabajar con el IDE que más os guste

Espero haber animado a usar este framework que cada día se utiliza más en aplicaciones, y ofrece características como la asincronía o la programación reactiva que no deja indiferente a la hora de poner aplicaciones en producción con *altas cargas de peticiones*.

Aquí os dejo un pom.xml de prueba

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.1.8.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository
-->
  </parent>
  <groupId>com.example</groupId>
  <artifactId>demo</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>demo</name>
  <description>Demo project for Spring Boot</description>

  <properties>
    <java.version>1.8</java.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-
elasticsearch</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-
security</artifactId>
    </dependency>
```

```
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-
thymeleaf</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-
webflux</artifactId>
        </dependency>

        <dependency>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
            <optional>true</optional>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-
test</artifactId>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>io.projectreactor</groupId>
            <artifactId>reactor-test</artifactId>
            <scope>test</scope>
        </dependency>
        <dependency>
<groupId>org.springframework.security</groupId>
            <artifactId>spring-security-
test</artifactId>
            <scope>test</scope>
```

```
                </dependency>
            </dependencies>

            <build>
                <plugins>
                    <plugin>
<groupId>org.springframework.boot</groupId>
                                <artifactId>spring-boot-maven-
plugin</artifactId>
                    </plugin>
                </plugins>
            </build>

</project>
```

Saludos